What Makes Financial Tasks Hard for Frontier LLMs? A Cross-Model Cross-Benchmark Analysis

Anonymous ACL submission

Abstract

Frontier large language models (LLMs) have reached near-saturation on mathematics and coding benchmarks, yet remain far from saturation on financial question tasks. To identify why this gap persists, we present a cross-benchmark, cross-model analysis of 3.4k financial tasks collected from five recent benchmarks evaluated with four frontier LLMs. We define a hard split, i.e., samples that all four models answer incorrectly, to isolate failures that reflect genuine task difficulty rather than idiosyncratic model weaknesses. Using this hard split, we analyze difficulty drivers across three intersecting dimensions: the financial concepts that categorize each task, the analytical operations required to solve it, and the error modes that frontier models produce when they fail. This analysis reveals that difficulty stems from a structured interaction between the financial concept and the analytical operation required. Moreover, LLM failures are convergent: models from different providers produce nearly identical error profiles, indicating shared architectural ceilings. Finally, dominant failure modes reside in document grounding and knowledge-boundary recognition, rather than computational operations.

1 Introduction

Large language models (LLMs) have achieved rapid gains on established reasoning benchmarks. On competition mathematics (e.g., HMMT (HMMT, 2025)), graduate-level science (e.g., GPQA (Rein et al., 2024)), and coding tasks (e.g., SWE-bench (Jimenez et al., 2024)), state-of-the-art performance has risen from below human baselines to near-saturation within two years (OpenAI, 2026; Anthropic, 2026a). These results have prompted claims that frontier models are approaching expert-level competence across technical domains (Sajadieh et al., 2026).

Performance on financial tasks has not followed this trajectory. Fig. 1 displays benchmark performance over successive model generations, comparing frontier math and financial tasks. On HMMT 25 (HMMT, 2025), scores rose from 30% to near-100% saturation between GPT-4.1 and Claude Opus 4.6. Conversely, financial benchmarks present a significantly less saturated different curve: for frontier performance on PRBench Finance (Akyürek et al., 2025), GPT-4.1 also started around 34%, but it only rose to 52% by Opus 4.6. These gains are positive but far smaller in magnitude and nowhere near saturation. The gap is widening in relative terms: Going from GPT-4 to GPT-5 closed roughly 50% of the remaining headroom on HMMT 25, but only around 10% on financial tasks. This divergence suggests that the scaling and architectural advances driving progress on clean, self-contained reasoning problems do not transfer proportionally to the financial domain. In this work, we investigate why this gap persists and what specific capabilities it exposes. Understanding the source of difficulty requires more than reporting aggregate scores: one must decompose failures by the financial concept involved, the analytical operation required, and their interaction, while also determining whether models fail via shared mechanisms or exhibit distinct error patterns. In contrast, existing studies evaluate models on individual financial benchmarks in isolation, reporting overall accuracy without distinguishing between failure types or difficulty drivers (Reddy

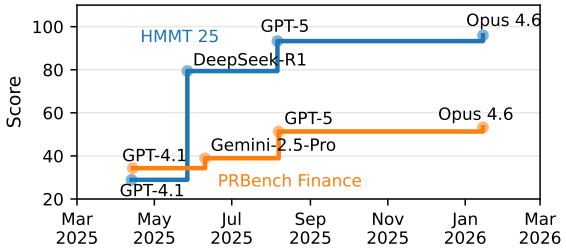

Figure 1: Frontier performance trends: math vs. finance.

et al., 2024; Chen et al., 2021). A cross-benchmark, cross-model analysis is therefore needed to disentangle these factors and identify whether the bottleneck is fundamental to financial reasoning or an artifact of individual benchmark design.

We aim to find the causes for why financial reasoning has been harder to saturate compared to reasoning problems targeted by frontier models. Intuitively, this may happen for a variety of reasons that compound within individual questions. Unlike mathematics, where problem statements are self-contained and notation is standardized, financial disclosures vary by company, jurisdiction, and reporting period. Many questions further require chaining arithmetic operations over values that must first be correctly retrieved, coupling retrieval difficulty with reasoning difficulty. Domain-specific conventions, such as accounting standards, unit scaling, and sign conventions, introduce systematic error modes that are less prevalent in mathematical reasoning. Finally, tasks such as financial forecasting and rubric-based advisory evaluation require open-ended professional judgment that resists reduction to a single extractable answer.

To test these hypotheses empirically, we evaluate four frontier LLMs—Claude Opus 4.6, Claude Sonnet 4.6, GPT-5.4, and Nemotron 3 Super 120B—across five financial benchmarks: FinQA (Chen et al., 2021), DocFinQA (Reddy et al., 2024), FinanceQA (Mateega et al., 2025), PRBench Finance (Akyürek et al., 2025), and SECQUE (Yoash et al., 2025). We find that around 26% of questions defeat every model tested, and the best single model achieves only 64% average accuracy. We address the decomposition gap through a multi-layer taxonomy analysis. We first perform a difficulty decomposition where each prompt is labeled along three taxonomies: financial concepts, analytical operations and error modes. These are then used to identify specific intersections where frontier LLMs succeed or fail. Our contributions are as follows:

- We establish the first cross-model, cross-benchmark difficulty analysis for financial tasks, defining hard/easy splits by model consensus.
- We introduce a multi-layer task taxonomy over concepts, operations, and error modes that decomposes problem hardness into interpretable factors, revealing interaction effects that drive difficulty.
- We observe that model failures largely partition into two regimes: document-grounding errors for quantitative tasks and knowledge-boundary fabrication for qualitative tasks. Error profiles converge across models, indicating shared limitations that scaling alone cannot resolve and motivating targeted intervention design.

2 Related Work

Recent years have seen a proliferation of benchmarks targeting numerical reasoning over financial documents. FinQA (Chen et al., 2021) introduced the task of generating executable programs from earnings-report tables and text, establishing that even purpose-built models struggle with multi-step arithmetic grounded in semi-structured financial data. DocFinQA (Reddy et al., 2024) extended this paradigm to full-length 10-K filings, demonstrating that long-context grounding is a distinct and compounding difficulty axis. FinanceQA (Mateega et al., 2025) contributed expert-authored questions requiring professional judgment and assumption-making, including tasks with no single extractable gold answer, which pushes evaluation beyond formulaic computation. FinanceBench (Islam et al., 2023) similarly targets real-world financial analysis but focuses on shorter, factoid-style questions over corporate filings. FinDER (Choi et al., 2025) addresses financial entity and relation extraction, complementing QA benchmarks with structured information extraction tasks. These evaluations typically report aggregate accuracy on individual benchmarks without decomposing failures by task structure or comparing error mechanisms across benchmarks

A parallel line of work evaluates financial reasoning in more complex, multi-turn settings. PRBench (Akyürek et al., 2025) introduced rubric-based professional evaluation with 10–30 expert-curated criteria per question, shifting the focus from single-answer correctness to completeness and professional rigor. SECQUE (Yoash et al., 2025) targets SEC-filing comprehension with questions requiring synthesis across regulatory disclosures. The Finance Agent benchmark (Bigeard et al., 2025) tests agentic workflows requiring tool use and multi-step retrieval. Finally, several recent efforts have focused on failure modes in the context of trustworthiness, such as: PHANTOM, which targets financial QA hallucination (Ji et al., 2025); FinBias, which targets factuality and fairness in finance (Mehrotra et al., 2025); and CNFinBench, which targets safety and compliance in Chinese

finance (Ding et al., 2025). Our work complements these efforts by providing a fine-grained error typology that distinguishes failure modes at a more mechanistic level.

3 Characterizing the Hardness Landscape of Financial Benchmarks

3.1 Frontier LLM Performance

We conduct our experiments across a collection of financial benchmarks, including DocFinQA, FinanceQA, FinQA, PRBench Finance, and SECQUE; see Appendix A.1 and Tab. 2 for further details. We conduct our analysis over four frontier LLMs across all financial benchmarks: Claude Sonnet 4.6 (Anthropic, 2026b), Claude Opus 4.6 (Anthropic, 2026a), Nemotron 3 Super 120B (NVIDIA, 2026), and GPT-5.4 (OpenAI, 2026). These models provide a representative set of recent model releases from frontier AI labs with a 1M token context window.

We provide a breakdown of benchmark performance by model in Table 1. Using weighted-average scores (weighted by benchmark sample count), the overall performance ranges from approximately 52% for Claude Sonnet 4.6 to 64% for Claude Opus 4.6. However, the per-benchmark breakdown reveals notable observations. Claude Opus 4.6 is consistently the best performing model, leading on four of five benchmarks. Its advantage is most pronounced on DocFinQA and SECQUE, and it holds a meaningful edge on PRBench Finance and FinanceQA. On the other hand, Claude Sonnet 4.6 achieves strong scores on FinQA, SECQUE, and PRBench (generally second only to Opus 4.6), but collapses to 13.9% on DocFinQA, considerably lower than the other models. Finally, GPT 5.4 and Nemotron 3 Super 120B perform similarly on average, and trade positions depending on the financial benchmark. GPT-5.4 dominates on SECQUE and FinanceQA, while Nemotron edges ahead on DocFinQA and PRBench Finance.

These results reveal that financial tasks remain difficult, as no benchmark is fully performance-saturated by any model, and the average scores hover between 50-60% accuracy. However, it remains to be seen what exactly drives difficulty in financial tasks. From Tab. 1 alone, it is difficult to disentangle genuine task difficulty from idiosyncratic model weaknesses; additionally, aggregate scores also conflate benchmark-specific design choices such as document length and evaluation rubric with genuine financial reasoning difficulty. This motivates a cross-benchmark difficulty analysis where we classify each sample across all benchmarks by the number of models that answered it incorrectly. Counting how many models fail on each sample provides a design-agnostic difficulty signal that cuts across these evaluation paradigms.

3.2 Difficulty Analysis Methodology

To establish a measure of question difficulty, we adopt a consensus-based difficulty scoring approach. On the 3.4k evaluation samples spanning all datasets, we count how many of the 4 models answered each question correctly. For benchmarks with binary correctness (FinanceQA, FinQA Original, DocFinQA), a sample is correct if the model's answer exactly matches the gold answer. For PRBench Finance, which uses a continuous multi-rubric score in [0, 1], we threshold at 0.5 to binarize correctness. SECQUE samples are considered correct when the LLM judge panel decision is fully correct, and incorrect otherwise.

For each sample, we define three difficulty tiers: (i) *easy*, where all models are correct; (ii) *hard*, where all models are wrong; (iii) *medium*, where at least 1 model was wrong and 1 model was correct. This strict definition, which requires unanimous failure for "hard" and unanimous success for "easy", ensures that the hard set captures questions that are universally difficult for the current generation of frontier models, rather than questions that a single weak model might miss.

3.3 Difficulty-Categorized Statistics

The overall split by difficulty across all samples is shown in Fig. 2. Around 26% of samples defeat every frontier model tested. This stands in stark contrast to recent reports on general-purpose benchmarks, e.g., MMLU, where frontier models achieve 85–95% accuracy (Vals AI, 2026). The financial domain presents a qualitatively different challenge, where a significant amount of problems yield poor performance.

Across benchmarks, the difficulty distribution varies markedly; clearly, DocFinQA, FinanceQA, and PRBench Finance exhibit higher difficulty than FinQA and SECQUE. DocFinQA stands out as the most consistently difficult benchmark: while its hard rate (33.2%) is the third highest, its easy rate is by far the lowest (3.4%), meaning nearly all of its samples trip up at least one frontier model. This aligns with the benchmark requiring reasoning over

Table 1: Benchmark performance by model.

Model	Weighted Avg.	DocFinQA	FinanceQA	FinQA	PRBench	SECQUE
Claude Opus 4.6	63.97	53.36	47.97	78.81	50.26	69.91
GPT 5.4	55.55	34.49	43.24	77.68	39.61	65.13
Nemotron 3 Super 120B	52.18	40.56	31.08	72.80	40.27	47.43
Claude Sonnet 4.6	51.68	13.88	45.95	77.86	46.30	67.43

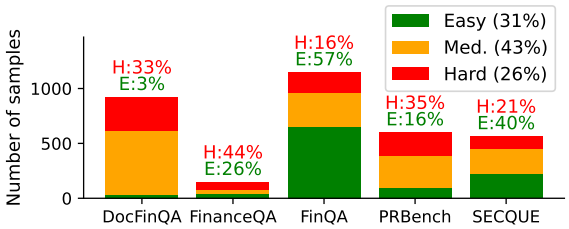

Figure 2: Benchmark difficulty breakdown. Across all benchmarks, roughly 26% are hard, 43% are medium, and 31% are easy. Per-benchmark hard/easy rates are displayed on top of each bar.

long financial documents. On the other hand, FinanceQA has the highest hard rate (43.9%) but also a substantial easy share (26.4%), indicating a bimodal distribution where questions tend to be either straightforward or very challenging, with comparatively few samples in the intermediate zone (29.7% medium, versus 63.4% for DocFinQA). FinQA and SECQUE are the easiest of the benchmarks used. FinQA has a mean of 1.25 models wrong; more than half its samples are solved by all four models. SECQUE shows 40.2% all-correct with a mean of 1.50 wrong. Both benchmarks contain a significant proportion of questions involving simple arithmetic or factual retrieval, which current LLMs handle reliably (Sajadieh et al., 2026).

3.4 Discussion

The existence of a difficult subset of significant size that takes up 26% of all samples where each frontier model fails demonstrates that financial tasks poses challenges that are fundamentally underserved by current model capabilities. The high inter-model agreement on the hard set suggests that the bottleneck likely lies in structural properties of the questions and documents themselves: the type of reasoning required, how information is distributed within financial filings, and how models handle numerical extraction from dense tabular contexts. The subsequent sections investigate these cross-benchmark dimensions directly by utilizing the hard, medium, and easy classification to discern the the underlying limitations preventing better financial task performance in current LLMs.

4 Multi-Layer Task Taxonomy

Now that we have established that a significant percentage of tasks across recent benchmarks cannot be solved by frontier LLMs, we seek to understand the different aspects of these tasks that cause LLMs to fail. This motivates the following question: when frontier LLMs fail on financial tasks, do they fail because the financial concept is hard, because the required operations to solve the task are hard, because certain error modes recur, or because of interactions among all three? To answer this, we develop a multi-layer task taxonomy over *financial concepts*, *analytical operations*, and *model error modes*. Each financial task is taxonomized according to these three angles in order to draw trends and observe failure tendencies in current LLMs.

4.1 Financial Concept Taxonomy

We begin by classifying each task across all benchmarks into 21 concept categories, which are listed in Appendix A.2, Tab. 3; these concepts are a superset of the financial topics defined in PRBench Finance. We applied LLM-assisted labeling to all question texts and gold responses across all datasets using GPT 5.4 with structured output.

Fig. 3 shows the concept-taxonomized tasks ordered by hard rate, i.e., the percentage of tasks within a concept where all models were incorrect. It can be seen that concepts which are more quantitative in nature, such as equity & dividends and revenue & margins are relatively easier. On the other hand, concepts involving qualitative advice or requiring qualitative assessments and/or recommendations, such as alternatives & private markets and wealth management, rank on the harder side.

We additionally report the enrichment score ES, where for a concept c,

$$\mathsf{ES}(c) = \frac{\text{fraction of hard questions falling in concept } c}{\text{fraction of all questions falling in concept } c}.$$

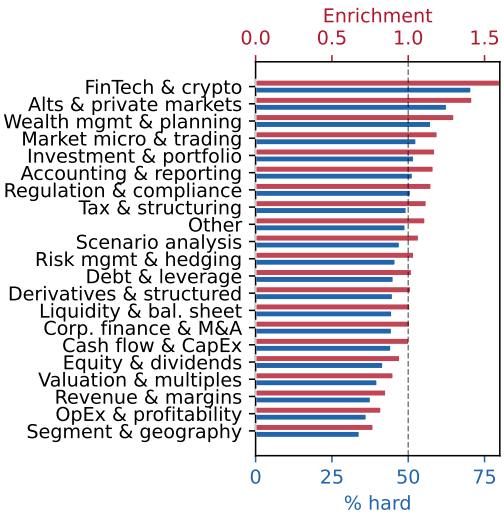

Figure 3: Hard rate and enrichment score of financial tasks taxonomized by financial concept.

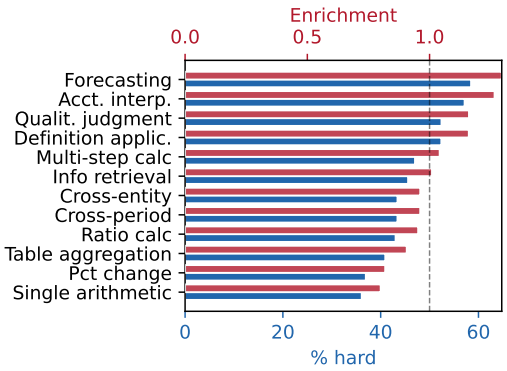

Figure 4: Hard rate and enrichment score of financial tasks taxonomized by analytical operation.

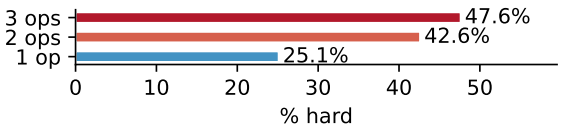

Figure 5: Hard rate increases with number of operations.

ES accounts for the imbalance of samples counts across concepts. While hard rate reports the raw ratio of hard samples within a concept, enrichment measures how overrepresented a concept is among hard questions relative to how often the concept appears in the dataset; ES > 1 implies that a concept is overrepresented among hard questions, whereas ES < 1 implies underrepresentation. It can be seen that the ranking of all concepts by enrichment remains the same as before, indicating the hard rate is not so susceptible to sample imbalance.

4.2 Analytical Operation Taxonomy

Next, we assign various analytical operations (out of 12 total) to each task; these are listed in Tab. 4. We again apply LLM-assisted labeling using GPT-5.4 to all the prompts and gold responses, but allow GPT-5.4 to assign a varying number of 1-3 operations to each financial task, since tasks may require different combinations of operations to solve the task. Fig. 4 shows the hard rate and enrichment score according to the operation taxonomy. Here, the bars represent the hard rate and enrichment across all instances where each operation was needed, since each sample may have multiple operations assigned. Similar to the concept taxonomy, operation sample imbalance does not play a significant role in the ranking of operations by difficulty. Furthermore, it can be seen that simpler, quantitative operations, like percentage change and single arithmetic are easier, whereas tasks requiring domain-specific terminology and qualitative judgment rank higher in terms of difficulty. This correlates with the concept difficulty ranking in Fig. 3; more detailed cross-taxonomy analyses can be found later in Sec. 5.

Aggregating all financial tasks by the number of operations in Fig. 4 shows that the hard rate increases with the number of operations. However, while there is a significant jump in hard rate going from 1 to 2 operations, there is less of an increase going to 3 operations, indicating that the types of operations involved play a more significant role regarding problem difficulty compared to the number of operations. Furthermore, aggregating hard rates by operation combination, shown in Fig. 6, displays how tasks with different combinations of operations vary by difficulty. For the most part, marginally easy operations that are combined with eachother have low hard rates, and vice versa for marginally hard operations. When marginally easy/hard operations are cross-combined, however, there is more variability: qualitative judgment combined with percentage change is difficult, but multi-step calculation combined with percentage change remains easy. This further supports that purely mathematical operations, even if they are more complex, can be readily handled by frontier LLMs, but mathematical operations combined with qualitative operations remain challenging.

4.3 Error Mode Taxonomy

Our final taxonomy categorizes the errors that different LLMs make. Whereas the concept/operation taxonomies were performed across all samples, the error taxonomy is performed on the hard split. Each

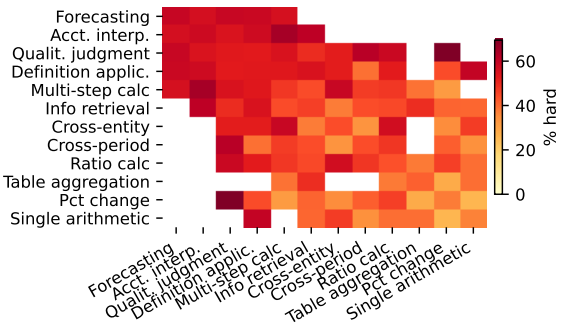

Figure 6: Hard rate by operation combination.

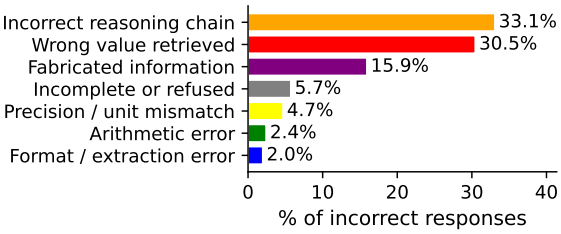

Figure 7: Error type distribution of model responses.

prompt/response pair from each model, along with the gold response, is taxonomized by GPT-5.4; the full error taxonomy is delineated in Tab. 5. The error type distribution across the hard split is shown in Fig. 7. It can be seen that incorrect reasoning and information retrieval account for nearly 70% of the error types; fabricated information ranks third at around 17%. The remaining error types, such as arithmetic and format errors, are comparatively infrequent. The error type breakdown by model across all benchmarks is shown on the left side of Fig. 8; a more detailed view for each individual benchmark can be found in Appendix B, Fig. 11. The distribution of error types do not differ significantly across all 4 models, indicating that these frontier models fail in similar ways. Nemotron 3 tends to have less information retrieval and reasoning errors compared to other models, but have a higher percentage of unit mismatches and incomplete responses. On the other hand, GPT-5.4 appears to have more information retrieval errors, but fabricates information less frequently.

The error type breakdown by benchmark, shown on the right side of Fig. 8, shows that error profiles are largely determined by task design rather than model capability. DocFinQA and FinanceQA are majority characterized by retrieval errors; both benchmarks require models to locate specific numerical facts within structured financial documents. FinanceQA additionally shows a substantial reasoning component, reflecting its multi-hop calculation questions. PRBench Finance presents a significant fabrication component, with limited retrieval errors. SECQUE distributes errors across reasoning, incomplete/refused, and wrong value retrieved.

5 Cross-Taxonomy Difficulty Analysis

In this section, we analyze across taxonomies to identify precise difficulty drivers behind financial tasks and deepen the understanding of frontier LLMs' strengths and weaknesses. We first analyze how concepts, operations, and error modes relate to eachother, before presenting example case studies (Appendix B) that concretely highlight notable ways in which frontier models succeed or fail. All enumerated concepts/operations follow the ordering by marginal hard rates found in Sec. 4.

5.1 Task Structure vs. Error Modes

We have assigned each financial task a difficulty rating, a financial concept, 1-3 operations, and an error. We now use this metadata to visualize their relationships.

Operations/concepts × errors. Given an operation or a concept, we report the distribution over error types in Fig. 9, where samples are model responses aggregated over the hard split. For example, the first concept row in this heatmap reports the error type distribution for all hard sample responses where the financial task primarily involved FinTech. Likewise, the first operation row in this heatmap reports the distribution of error types for all hard sample responses involving the information retrieval operation.

For many of the quantitative concepts and operations, such as the revenue, equity, and liquidity concepts and the ratio calculation, percentage change operations, the error distribution concentrates on wrong value retrieved and incorrect reasoning chain. Arithmetic errors almost never happen. For concepts, many of the qualitative tasks, such as FinTech, regulation, and wealth and risk management have high occurrences of fabricated information. This is supported by the qualitative judgment operation landing a majority share of errors in fabrication. Additionally, the fabrication is common among the forecasting operation; this is consistent with high fabrication error rates on concepts that involve forecasting, such as trading, risk management & hedging, and scenario analysis.

Three-way interactions: Operations × concepts × errors. In Fig. 10, we report the hard rate broken

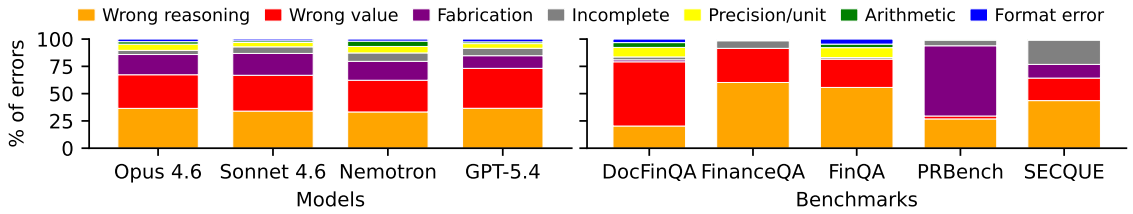

Figure 8: Error type breakdown by models and benchmarks.

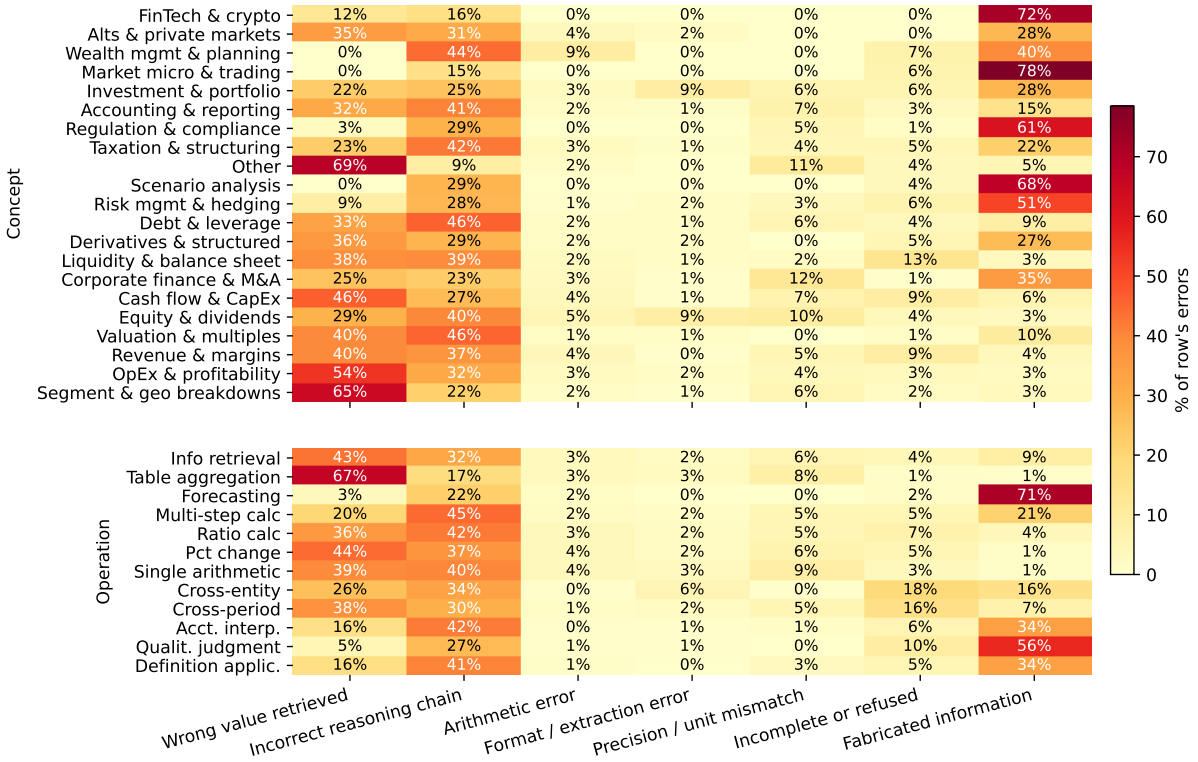

Figure 9: Distribution of error types for each concept/operation, aggregated over hard split.

down by concept-operation pairs; Tab. 6 displays a summary table highlighting the top operations and error modes for each concept. As can be seen, most of the hardest concepts primarily involve qualitative judgments and intrinsically knowing relevant definitions. These primarily result in fabrication errors. For the remaining concepts, most involve information retrieval and result in errors of wrong value retrieved and incorrect reasoning. Of these, the harder concepts in the middle of the list tend to have qualitative judgment as a secondary operation, whereas the easier concepts at the bottom involve purely quantitative operations like ratio calculation.

5.2 Key Takeaways

The joint analysis of financial concepts, analytical operations, and error classifications reveals several non-obvious patterns about where and why frontier LLMs fail on financial tasks.

Dense tables hurt grounding but not reasoning. *Incorrect reasoning* is the leading error category overall (∼33%, Fig. 7), yet Fig. 9 shows *wrong value retrieved* (∼31%) dominates in concepts backed by dense tabular data—balance sheets (39%), cash flow (46%), revenue (40%)—and table aggregation operations (67%). Within these same concepts, Fig. 10 shows that routine quantitative operations (ratio calculation, percentage change) have hard rates of only 18–30%, while *information retrieval / lookup* exceeds 31–48%. Reasoning works well when grounded in the right inputs; the primary constraint is attention over long tables with repeated column headers and nested line items. See case studies in Appendix B.1.

Fabrication concentrates when ground truth is sparse. Some of the hardest concepts share a common property in that their questions ask about details or require the model to infer content not explic-

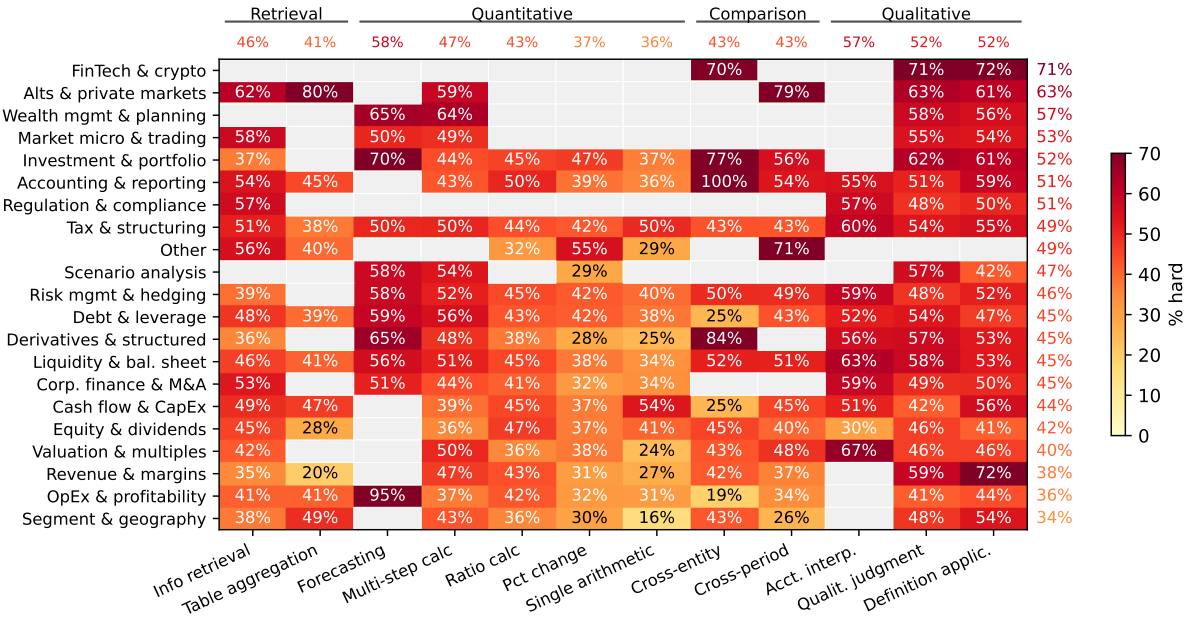

Figure 10: Hard rate for each concept-operation pair; concepts on y-axis and operations on x-axis.

itly stated in the source document, resulting in fabricated information: market microstructure (78% fabrication; Fig. 9), FinTech (72%), scenario analysis (68%), and regulation (61%). Case studies in Appendix B.2 show these error modes for scenario analysis and regulation tasks. Risk management (51%), corporate finance (35%), and investment portfolio (28%) exhibit the same pattern at lower but elevated rates. In contrast, fabrication is nearly absent (<5%) for revenue, operating expenses, segments, and equity, which are topics where the answer is a number explicitly present in a table or elsewhere in the context provided.

Difficulty follows a two-regime structure determined by operations. The hardest concepts, e.g., FinTech (71% hard rate; Fig. 3), alternatives (63%), wealth management (58%), primarily require the operations of *qualitative judgment* and *definition application*, and their dominant error mode is fabrication. After that, concepts such as debt, equity, cash flow, revenue, and segments (34–45% hard) rely on the *information retrieval* operation paired with other quantitative operations—e.g., ratio calculation, percentage change, multi-step arithmetic; see Tab. 6. Their errors are wrong-value-retrieved and incorrect reasoning. A transition zone (accounting 52%, regulation 51%, taxation 50%) marks the boundary: these concepts use information retrieval as the primary operation but retain qualitative judgment as a secondary one, producing a mixed error profile. Difficulty increases when tasks transition from find-and-compute to judge-and-synthesize.

LLM error modes are convergent across models. Across benchmarks, the four models produce very similar error type distributions in Fig. 8. Additionally, Fig. 11 reveals that this remains true within each benchmark: information retrieval errors dominate all models within DocFinQA; reasoning errors dominate all models in FinQA; and SECQUE distributes errors across reasoning, information retrieval, fabrication, and incompleteness similarly. Case studies in Appendix B.3 illustrate this phenomenon for several tasks in which all models make the same mistake. This indicates structural limitations of the current transformer paradigm, pretraining and finetuning strategies rather than individual model weaknesses. Further scaling within the existing architecture is unlikely to resolve these failure modes; they require targeted interventions for the specific hardness trends described above.

6 Conclusion

Across multiple recent financial benchmarks and four frontier LLMs, we find that 26% of questions defeat every model, and draw trends across question types and failure modes. Difficulty is highly varied: specific concept/operation combinations yield extremely high failure rates, while others are essentially solved. Closing the financial performance gap requires better document grounding and finetuning on finance domains covering irregular disclosures such as M&A forecasting.

Limitations and Risks

Our difficulty scores are derived from only four recent LLMs, so the taxonomy may reflect shared architectural blind spots or stochastic variation rather than intrinsic difficulty. The five benchmarks are all English-language and predominantly U.S.-centric, so findings may not generalize to other languages, jurisdictions, or qualitative financial tasks. Finally, our analysis is a static snapshot; rapid model development may shift difficulty rankings with newer checkpoints. This work is an analytical study of existing public benchmarks, so we see limited direct risk.

References

Afra Feyza Akyürek, Advait Gosai, Chen Bo Calvin Zhang, Vipul Gupta, Jaehwan Jeong, Anisha Gunjal, Tahseen Rabbani, Maria Mazzone, David Randolph, Mohammad Mahmoudi Meymand, and 1 others. 2025. Prbench: Large-scale expert rubrics for evaluating high-stakes professional reasoning. *arXiv preprint arXiv:2511.11562*.

Anthropic. 2026a. Claude opus 4.6 system card. Technical report, Anthropic.

Anthropic. 2026b. Introducing Claude Sonnet 4.6. Accessed: 2026-04-29.

Antoine Bigeard, Langston Nashold, Rayan Krishnan, and Shirley Wu. 2025. Finance agent benchmark: Benchmarking llms on real-world financial research tasks. *arXiv preprint arXiv:2508.00828*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, and 1 others. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy-yong Sohn, and Alejandro Lopez-Lira. 2025. Finder: Financial dataset for question answering and evaluating retrieval-augmented generation. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pages 638–646.

Jinru Ding, Chao Ding, Wenrao Pang, Boyi Xiao, Zhiqiang Liu, Pengcheng Chen, Jiayuan Chen, Tiantian Yuan, Junming Guan, Yidong Jiang, and 1 others. 2025. Cnfinbench: A benchmark for safety and compliance of large language models in finance. *arXiv preprint arXiv:2512.09506*.

HMMT. 2025. Harvard-mit mathematics tournament. https://www.hmmt.org/.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. Financebench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*.

Lanlan Ji, Dominic Seyler, Gunkirat Kaur, Manjunath Hegde, Koustuv Dasgupta, and Bing Xiang. 2025. PHANTOM: A benchmark for hallucination detection in financial long-context QA. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, volume 2024, pages 54107–54157.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Spencer Mateega, Carlos Georgescu, and Danny Tang. 2025. Financeqa: a benchmark for evaluating financial analysis capabilities of large language models. *arXiv preprint arXiv:2501.18062*.

Shreshth Mehrotra, Raghavendra P, Balraj Prajesh, Hrishikesh Kambale, and Puspita Majumdar. 2025. *Unmasking Bias in Financial AI: A Robust Framework for Evaluating and Mitigating Hidden Biases in LLMs*, page 951–959. Association for Computing Machinery, New York, NY, USA.

NVIDIA. 2026. Nemotron 3 super: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2604.12374*.

OpenAI. 2026. Gpt-5.4 thinking system card - deployment safety hub.

Varshini Reddy, Rik Koncel-Kedziorski, Viet Dac Lai, Michael Krumdick, Charles Lovering, and Chris Tanner. 2024. Docfinqa: A long-context financial reasoning dataset. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 445–458.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Sha Sajadieh, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Lapo Santarlasci, Juan Pava, Nestor Maslej, Russ Altman, Erik Brynjolfsson, Carla Brodley, Jack Clark, Virginia Dignum, Vipin Kumar, James Landay, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, and 4 others.

2026. The AI index 2026 annual report. Technical report, AI Index Steering Committee, Institute for Human-Centered AI, Stanford University.

Vals AI. 2026. Vals ai benchmarks. https://www.vals.ai/benchmarks.

Noga Ben Yoash, Meni Brief, Oded Ovadia, Gil Shenderovitz, Moshik Mishaeli, Rachel Lemberg, and Eitam Sheetrit. 2025. Secque: A benchmark for evaluating real-world financial analysis capabilities. *arXiv preprint arXiv:2504.04596*.

A Experimental Details

A.1 Benchmarks

We first provide a summary table of all benchmarks in Tab. 2. Next, we provide high-level details on how each benchmark was evaluated. Each benchmark follows the original paper's description. Anthropic and OpenAI models were accessed via API call, and Nemotron 3 Super was accessed via local vLLM serving (Kwon et al., 2023) on 4 NVIDIA L40S GPUs.

DocFinQA. DocFinQA (Reddy et al., 2024) is evaluated as a program-of-thought task rather than with answer matching or LLM judging. For each example, the model is asked to answer the financial question by generating a Python program whose final line assigns the result to `answer`. The evaluator extracts and executes the program in a restricted environment, and if execution fails, it falls back to extracting a numerical answer from the free-form response. Performance is measured with execution accuracy using tolerant numerical comparison so that small formatting or rounding differences do not count as errors.

FinQA. FinQA (Chen et al., 2021) is evaluated in the same general way as DocFinQA. The model is given a financial report excerpt constructed from text and table content and is instructed to answer by writing a Python program with the final result stored in `answer`. The evaluator extracts and executes the program, with fallback to numerical extraction from plain text when necessary. Final scoring is based on execution accuracy with tolerant numerical matching.

FinanceQA. FinanceQA (Mateega et al., 2025) is evaluated with an LLM-as-a-judge setup. The model receives the question, along with any provided context, and produces an open-ended answer. A separate judge model then compares the response against the reference answer and returns a structured binary correctness decision. The judging rubric requires agreement on the final numerical value or qualitative conclusion, while allowing only minor rounding differences and not allowing materially different assumptions.

PRBench. PRBench (Akyürek et al., 2025) is evaluated against expert-authored rubrics rather than a single gold answer. Each example is formatted as a multi-turn conversation, and the model is scored only on its response to the final user

Benchmark	n	Description
FinQA (Chen et al., 2021)	1,147	Program-generation questions over earnings-report tables and text; requires multi-step arithmetic grounded in semi-structured data.
DocFinQA (Reddy et al., 2024)	922	Questions over full-length 10-K filings; tests long-context grounding with dense tabular disclosures.
PRBench (Akyürek et al., 2025)	600	Rubric-based professional evaluation with 10–30 expert criteria per question; measures completeness and professional judgment.
SECQUE (Yoash et al., 2025)	565	SEC-filing comprehension requiring synthesis across regulatory disclosures and cross-document reasoning.
FinanceQA (Mateega et al., 2025)	148	Expert-authored questions requiring professional judgment and assumption-making.

Table 2: Benchmarks used in this study. n denotes the number of questions evaluated per benchmark.

turn. A judge model evaluates the response independently against each rubric criterion, and these criterion-level decisions are combined using the rubric's weights. The final sample-level score is a normalized weighted rubric score, and overall performance is the mean of these scores across examples.

SECQUE. SECQUE (Yoash et al., 2025) uses open-ended answering followed by panel-style LLM judging. After the model answers a question based on an SEC filing excerpt, the response is judged five times, with each judgment assigning a score of 0, 1, or 2 for incorrect, partially correct, or fully correct. These five scores are summed and converted into a final sample-level label using fixed thresholds, where only responses with a high enough aggregate are treated as fully correct. Results are then reported both as strict accuracy, which counts only samples whose aggregated label is fully correct, and as normalized accuracy, which gives partial credit to partially correct responses.

A.2 Multi-Layer Taxonomy

Both the concept / operation labeling and the error classification use GPT-5.4 at temperature 0 with Pydantic-enforced structured output, ensuring every API response conforms to a fixed JSON schema. Parts of this process were developed with the assistance of AI tools such as Claude Code and Copilot.

Concept and Operation Taxonomy The complete financial concept and analytical operation taxonomies are shown in Tabs. 3, 4. All 3,382 unique questions across the five benchmarks are labeled individually. Each call receives: (i) the question text, (ii) the ground-truth reasoning chain where available, and (iii) the gold answer. Providing solution methodology alongside the question enables the classifier to identify the required operations with high fidelity rather than guessing from the question text alone.

The structured-output schema returns: a primary financial concept, an optional secondary concept, 1–3 analytical operations (ordered by relevance), evidence type, answer type, boolean flags for multi-step math and retrieval, a confidence score, and a one-sentence rationale. Across all 3,382 labels, the mean self-reported confidence is 0.92 (median 0.95); only 53 questions (1.6%) fall below 0.7.

The system prompt is reproduced below (concept and operation lists abbreviated; full lists appear in Tabs. 3 and 4).

```
You are an expert financial analyst.
Your task is to classify a financial
question according to two taxonomies.

FINANCIAL CONCEPT TAXONOMY (choose
exactly one primary, one secondary or
'None'):
- Revenue, cost structure, and margins
- Operating expenses and profitability
- . . . [21 categories; see Table 3]

ANALYTICAL OPERATION TAXONOMY (choose
1-3 operations, ordered by relevance):
- Information retrieval / lookup
- Single arithmetic operation
- . . . [12 categories; see Table 4]

You will receive the question text, and
when available:
- A "Solution methodology" showing the
reasoning steps or program used to
solve the question.
- A "Ground-truth answer" with the
correct final answer.

Use ALL available information
(question, solution methodology,
and answer) to make the most
accurate classification. The solution
methodology is especially useful for
determining analytical_operations and
requires_multi_step_math.

For the question, determine:
1. finance_concept_primary: The main
financial concept category
2. finance_concept_secondary: A
secondary concept if the question spans
two areas, otherwise "None"
```

Table 3: Financial concepts and descriptions.

Finance Concept	Description
Revenue, cost structure, and margins	Analysis of sales generation, cost drivers, gross margins, contribution margins, and how changes in pricing, volume, or input costs affect profitability.
Operating expenses and profitability	Assessment of SG&A, R&D, compensation, overhead, operating leverage, EBITDA, EBIT, net income, and overall earnings performance.
Cash flow and capital expenditure	Evaluation of operating cash flow, free cash flow, capital spending, maintenance versus growth capex, cash conversion, and reinvestment needs.
Debt, leverage, and credit risk	Analysis of debt levels, leverage ratios, interest coverage, refinancing risk, credit ratings, default probability, covenants, and overall creditworthiness.
Liquidity, working capital, and balance sheet	Assessment of cash resources, short-term obligations, receivables, inventory, payables, current ratios, working capital efficiency, and balance sheet strength.
Equity, dividends, and shareholder returns	Analysis of equity issuance, buybacks, dividends, payout ratios, dilution, total shareholder return, and capital return policies.
Valuation, multiples, and market metrics	Use of valuation approaches such as DCF, comparable company analysis, precedent transactions, P/E, EV/EBITDA, price-to-book, and market capitalization metrics.
Corporate finance and M&A	Evaluation of capital allocation, acquisitions, divestitures, merger rationale, synergies, deal financing, accretion/dilution, and transaction structuring.
Investment strategy and portfolio analysis	Assessment of asset allocation, security selection, portfolio construction, diversification, risk-adjusted returns, benchmarks, and investment theses.
Taxation and cross-border structuring	Analysis of tax rates, deferred taxes, transfer pricing, jurisdictional tax exposure, repatriation, treaty considerations, and international structuring.
Accounting policies and financial reporting	Review of accounting standards, revenue recognition, lease accounting, impairment, consolidation, disclosure quality, earnings quality, and financial statement presentation.
Risk management, hedging, and stress testing	Evaluation of financial, market, credit, operational, and liquidity risks, including hedging programs, stress scenarios, VaR, and risk mitigation practices.
Derivatives and structured products	Analysis of options, futures, swaps, forwards, structured notes, securitized products, payoff profiles, embedded risks, and valuation considerations.
Market microstructure, trading, and liquidity	Assessment of trading venues, bid-ask spreads, order books, execution quality, market depth, transaction costs, and liquidity conditions.
Regulation, compliance, and ethics	Review of regulatory obligations, compliance controls, conduct risk, disclosure requirements, fiduciary duties, conflicts of interest, and ethical considerations.
Wealth management and financial planning	Analysis of client goals, asset allocation, retirement planning, estate planning, tax-aware investing, risk tolerance, and personalized financial advice.
Alternative investments and private markets	Assessment of private equity, private credit, hedge funds, real estate, infrastructure, venture capital, fund structures, illiquidity, and performance measurement.
Forecasting, scenario analysis, and sensitivity	Development of financial projections, downside/base/upside cases, key assumptions, sensitivities, scenario trees, and uncertainty analysis.
Segment, geography, and product breakdowns	Analysis of revenue, margins, growth, and profitability across business lines, regions, customer groups, or product categories.
FinTech, crypto, and digital assets	Assessment of digital finance, blockchain, crypto assets, tokenization, payments innovation, digital wallets, decentralized finance, and related regulatory risks.
Other / unclear	Used when the finance topic does not fit clearly into the listed categories or when the relevant concept is ambiguous or insufficiently specified.

```
3. analytical_operations: 1-3 analytical operations required to answer (most relevant first). Use 1 if a single operation suffices; use 2-3 only if the question genuinely requires chaining multiple reasoning types.
4. evidence_type: What evidence is needed (single_value, multiple_values, table, narrative, mixed)
5. answer_type: The expected answer format (numeric, percentage, boolean, categorical, text)
6. requires_multi_step_math: Whether solving requires more than one arithmetic step
7. requires_retrieval: Whether the answer requires locating specific information in a document
8. confidence: Your confidence in this classification (0.0 to 1.0)
```

Table 4: Analytical operations and descriptions.

Operation	Description
Information retrieval / lookup	Finding a specific financial fact, figure, disclosure, date, policy, metric, or statement from a source document, database, filing, transcript, or table.
Single arithmetic operation	Performing one basic calculation, such as addition, subtraction, multiplication, or division, using explicitly provided financial values.
Percentage change	Calculating the relative increase or decrease between two values, often across time periods, using the formula $\frac{\text{New Value}-\text{Old Value}}{\text{Old Value}} \times 100$.
Ratio calculation	Computing a financial ratio by dividing one metric by another, such as gross margin, debt-to-equity, current ratio, return on equity, or interest coverage.
Multi-step calculation	Solving a task that requires multiple sequential calculations, intermediate outputs, or formulas, such as estimating free cash flow, enterprise value, EPS accretion, or implied valuation.
Table aggregation	Summarizing, grouping, or combining values across rows, columns, segments, periods, geographies, or categories within a table.
Cross-period comparison	Comparing financial metrics across different reporting periods, such as quarter-over-quarter, year-over-year, trailing twelve months, or multi-year trend analysis.
Cross-entity comparison	Comparing financial performance, valuation, risk, capital structure, or operating metrics across companies, business units, sectors, funds, or comparable entities.
Definition application	Applying a stated financial, accounting, regulatory, or analytical definition to determine whether a metric, transaction, classification, or scenario meets the relevant criteria.
Accounting interpretation	Interpreting accounting treatment, financial statement classification, disclosure implications, recognition rules, measurement policies, or effects on reported results.
Forecasting or extrapolation	Projecting future financial values based on historical trends, assumptions, growth rates, run-rates, scenarios, or model inputs.
Qualitative financial judgment	Making a reasoned assessment that relies on financial context, business logic, risk evaluation, market conditions, management commentary, or professional judgment rather than calculation alone.

```
9. rationale_short: A brief
one-sentence explanation
```

The user message is constructed as:

```
Classify this financial question:
{question_text}
Solution methodology: {reasoning_chain}
Ground-truth answer: {gold_answer}
```

Error Mode Taxonomy The error taxonomy shown in Tab. 5. For the hard split (all four models incorrect), we classify each of the incorrect responses. Each API call receives: (i) the question text, (ii) the model's verbatim response, (iii) the correct answer including the ground-truth reasoning chain where available, and (iv) a source-document excerpt (first 3,000 words, providing sufficient context for retrieval and fabrication distinction without exceeding token budgets). The classifier assigns a primary error type, an optional secondary type, a brief evidence summary, and a confidence score. Mean confidence across all classifications is 0.96.

The system prompt is reproduced below.

```
You are an expert financial QA evaluator.
A model was asked a financial question
and answered INCORRECTLY. Your task
is to classify the type of error by
comparing the model's response against
the correct answer (including the
ground-truth reasoning chain where
available) and the source document.

You will be given:
1. The QUESTION
2. The MODEL RESPONSE (incorrect)
3. The CORRECT ANSWER - which may
include a solution methodology /
reasoning chain showing intermediate
steps, plus the final correct answer
4. A SOURCE DOCUMENT excerpt for
additional context

Use the ground-truth reasoning chain
to pinpoint exactly where the model
diverged from the correct approach.

ERROR TYPE TAXONOMY (choose exactly one
primary, one secondary or 'None'):
- Wrong value retrieved
- Arithmetic error
- Incorrect reasoning chain
- Fabricated information
- Precision / unit mismatch
- Incomplete or refused
- Format / extraction error

Definitions:
- Wrong value retrieved: The model
pulled the wrong number(s) from the
```

Table 5: Error taxonomy for model responses.

Error	Description	Examples
Wrong value retrieved	The model extracts or references an incorrect numerical value from the source document (e.g., wrong row, wrong year, wrong entity), but the downstream reasoning structure is plausible given the (incorrect) inputs.	Wrong row/year/entity selected; logic plausible given the wrong number
Arithmetic error	The model identifies the correct values but makes a computational mistake in applying the required operation (addition, division, percentage change, etc.).	Addition/division/percent-change computed incorrectly
Incorrect reasoning chain	The model applies a logically flawed approach: wrong formula, incorrect causal inference, or invalid multi-step logic, regardless of whether the retrieved values are correct.	Wrong formula; invalid multi-step logic; incorrect causal leap
Fabricated information	The response contains claims, figures, or assumptions not present in or derivable from the source document.	Hallucinated figures/assumptions; unsupported claims
Precision / unit mismatch	The model arrives at a numerically close answer but fails on unit conversion, rounding, scale (e.g., "in millions"), or sign convention.	Unit/scale mismatch ("millions"); rounding/precision issues; sign flipped
Incomplete or refused	The model declines to answer, produces an empty or truncated response, or provides only a partial answer that cannot be evaluated.	Refusal; blank/truncated; partial non-evaluable answer
Format / extraction error	The model's response cannot be parsed by the benchmark evaluator due to formatting issues (e.g., malformed code, extraneous text wrapping a numeric answer), even though the underlying answer may be correct.	Malformed code; extra text around numeric answer; unparsable output

```
document — e.g., wrong row, wrong
year, wrong entity, or confused
similar-looking values.
- Arithmetic error: The model identified
the correct input values but made
a calculation mistake (addition,
subtraction, division, percentage,
etc.).
- Incorrect reasoning chain: The model
used the right values and the right type
of formula, but applied wrong logic —
e.g., subtracted instead of divided,
used wrong time period comparison,
inverted numerator/denominator.
- Fabricated information: The model's
response contains facts, numbers, or
claims that are not present in the
source document.
- Precision / unit mismatch: The model's
approach and values are correct, but
the final answer has the wrong scale
(millions vs thousands), rounding, sign,
or unit convention.
- Incomplete or refused: The model
declined to answer, produced an empty
or irrelevant response, or gave a
partial answer that doesn't address the
question.
- Format / extraction error: The model's
reasoning appears correct, but the
final extracted answer is malformed,
truncated, or unparseable by the
evaluator.

Classify the error based on the PRIMARY
reason the answer is wrong. If multiple
error types apply, put the most
important one as primary and the second
as secondary.
```

The user message is constructed as:

```
QUESTION:
{question_text}

MODEL RESPONSE:
{model_response}

CORRECT ANSWER:
{gold_answer_with_reasoning}

SOURCE DOCUMENT (excerpt):
{document_first_3000_words}
```

B Additional Results

Fig. 11 shows the error type distribution over models broken down for each individual benchmark.

A summary table of concept, top operations and error modes can be found in Tab. 6. Below, we display error mode examples with concept and operation labels attached.

B.1 Case Studies: Information Retrieval

Case Study 1: Wrong Value Retrieved

Benchmark: DocFinQA **Model:** Sonnet 4.6
Concept: Cash flow & CapEx
Operation: Information retrieval / lookup
Question: what were total development costs in millions for the three year period?
Gold answer: Solution methodology / reasoning chain: table_row = [1899, 2546, 2786] # row labeled development costs incurred during the period a = sum(table_row)
Final correct answer: 7231
Model answer: "'python # Development costs for oil and gas producing activities from the Costs Incurred table # Values extracted from the document
Development costs by year (in millions) development_costs_2011 = 1882 # Total continuing operations development [. . .]
Why wrong: The model summed development costs

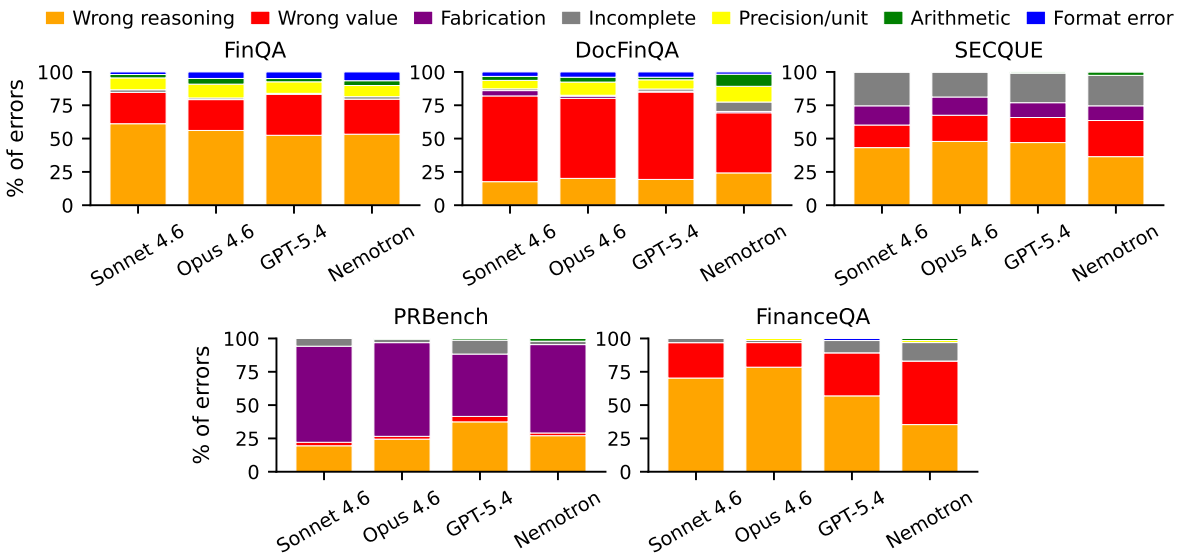

Figure 11: Error type distribution over models for each individual benchmark.

Concept	% Hard	Top Operations	Dominant Error Modes
FinTech & crypto	70.7	Qualit. judgment, Definition applic.	Fabricated information, Incorrect reasoning chain
Alts & private markets	62.7	Qualit. judgment, Definition applic.	Wrong value retrieved, Incorrect reasoning chain
Wealth mgmt & planning	57.5	Qualit. judgment, Definition applic.	Incorrect reasoning chain, Fabricated information
Market micro & trading	52.6	Qualit. judgment, Definition applic.	Fabricated information, Incorrect reasoning chain
Investment & portfolio	51.9	Qualit. judgment, Definition applic.	Fabricated information, Incorrect reasoning chain
Accounting & reporting	51.5	Info retrieval, Acct. interp.	Incorrect reasoning chain, Wrong value retrieved
Regulation & compliance	50.8	Info retrieval, Qualit. judgment	Fabricated information, Incorrect reasoning chain
Taxation & structuring	49.5	Info retrieval, Definition applic.	Incorrect reasoning chain, Wrong value retrieved
Other	49.1	Info retrieval, Pct change	Wrong value retrieved, Precision / unit mismatch
Scenario analysis	47.2	Forecasting, Qualit. judgment	Fabricated information, Incorrect reasoning chain
Risk mgmt & hedging	45.8	Qualit. judgment, Definition applic.	Fabricated information, Incorrect reasoning chain
Debt & leverage	45.2	Info retrieval, Ratio calc	Incorrect reasoning chain, Wrong value retrieved
Derivatives & structured	45.0	Definition applic., Qualit. judgment	Wrong value retrieved, Incorrect reasoning chain
Liquidity & balance sheet	44.7	Info retrieval, Ratio calc	Incorrect reasoning chain, Wrong value retrieved
Corporate finance & M&A	44.6	Qualit. judgment, Info retrieval	Fabricated information, Wrong value retrieved
Cash flow & CapEx	44.4	Info retrieval, Ratio calc	Wrong value retrieved, Incorrect reasoning chain
Equity & dividends	41.8	Info retrieval, Ratio calc	Incorrect reasoning chain, Wrong value retrieved
Valuation & multiples	39.9	Info retrieval, Multi-step calc	Incorrect reasoning chain, Wrong value retrieved
Revenue & margins	37.7	Info retrieval, Pct change	Wrong value retrieved, Incorrect reasoning chain
OpEx & profitability	36.4	Info retrieval, Pct change	Wrong value retrieved, Incorrect reasoning chain
Segment & geo breakdowns	34.1	Info retrieval, Ratio calc	Wrong value retrieved, Incorrect reasoning chain

Table 6: Summary table of concept, operation, and error mode relationship.

using 1,882, 2,537, and 2,769, but the correct table row values are 1,899, 2,546, and 2,786. Its total is therefore wrong primarily because it extracted the wrong figures from the document, though the resulting sum is also arithmetically based on those incorrect in [. . .]

Case Study 2: Wrong Value Retrieved

Benchmark: FinanceQA **Model:** Opus 4.6
Concept: Revenue & margins
Operation: Information retrieval / lookup
Question: What is Gross Profit in the year ending 2024?

Gold answer: Solution methodology / reasoning chain: Step 1: Identify the Formula The formula for Gross Profit is: Revenue - Cost of Goods Sold (COGS)
Step 2: Locate the Required Data I know both Revenue and COGS [. . .]
Model answer: ## Gross Profit Calculation for Fiscal Year 2024
Gross Profit is calculated as:
Gross Profit = Net Sales - Merchandise Costs
From the Consolidated Statements of Income: - **Net Sales (2024):** $249,625 million - **Merchandise Costs (2024):** $ [. . .]
Why wrong: The model used Net Sales of $249,625 million as revenue, but the correct gross profit calcu-

lation should use Total revenue of $254,453 million from the income statement. It therefore subtracted merchandise costs from the wrong top-line figure, producing $27,267 million instead of $32,095 million.

Case Study 3: Wrong Value Retrieved

Benchmark: DocFinQA **Model:** Sonnet 4.6
Concept: Liquidity & bal. sheet
Operation: Information retrieval / lookup
Question: what is the net change in the balance of accrual related to restructurings during 1999?
Gold answer: Solution methodology / reasoning chain: accrual_balance_1999 = 772 previous_accrual_balance = 8867 change = accrual_balance_1999 - previous_accrual_balance answer = change
Final correct answer: -8 [. . .]
Model answer: “‘python # Based on the financial document, we need to calculate the net change in the balance # of accrued restructuring charges during fiscal 1999
From the Consolidated Balance Sheets: # Accrued restructuring charges at December 3, 1999 accrued [. . .]
Why wrong: The model used 8,675 as the 1999 restructuring accrual balance, but the correct 1999 balance from the restructuring accrual rollforward is 772; 8,675 was the wrong figure retrieved from the document. After retrieving the wrong value, it then computed the change as 8,675 - 8,867 = -192 instead of the [. . .]

B.2 Case Studies: Fabrication

Case Study 1: Fabrication

Benchmark: PRBench Finance **Model:** GPT-5.4
Concept: Scenario analysis
Operation: Forecasting or extrapolation
Question: User: To implement the Weighted Deferred Revenue framework as a real-time monitoring tool, how can we ensure its predictive validity and scalability across time and peers? Can you please do a back-test WDR against past periods, like the slowdown in 2019 and the supply chain crunch in 2022, to see how early/effective it signaled a demand shift?
Gold answer: Solution methodology / reasoning chain: The response should present a concrete plan to test how effectively the Weighted Deferred Revenue (WDR) model predicts revenue or order trends. The response should specify required data inputs - such as quarterly deferred revenue, segment sales, and revenue mix. The response should detail the WDR computation [. . .]
Model response (excerpt): Yes — but first I need to correct and tighten the framework before back-testing it.
Important correction Some of the earlier Apple deferred revenue figures and the "85% services / 15% hardware" split were too strong as stated and should be treated as illustrative, not filing-verified. Apple does disclose contract liabilities / deferred revenue, but not a clean services-vs-hardware deferred revenue split. So if we want a **predictively valid, scalable** monitoring tool, we should use a framew [. . .]
Why wrong: The model admits key Apple deferred revenue figures and the 85% services / 15% hardware split were not filing-verified, then builds a new WDR 2.0 framework with invented weights and only directional, hypothetical back-test conclusions ('likely', 'should have'). The correct answer called for a concrete back-test methodology using defined inputs, normalization, event windows, and lead-lag validation [. . .]

Case Study 2: Fabrication

Benchmark: PRBench Finance **Model:** Nemotron 3 Super 120B
Concept: Regulation
Operation: Definition application
Question: User: Advise on whether Nigerian banks can prepay a loan to a foreign bank under a tier 2 loan agreement, highlighting CBN's policy on foreign borrowing. Additionally, state other regulatory capital requirements different from the above, if any.
Gold answer: State the CBN's policies on prepayment of loans by banks, which include a **prohibition on prepayment of loans**. Highlights CBN's policy on foreign borrowing, which includes that borrowings must be subordinated debts. Outline the features of Tier 2 capital under [. . .]
Model response (excerpt): ### Short Answer
Yes – Nigerian banks may prepay a Tier-2 loan to a foreign lender, provided they obtain prior written approval from the Central Bank of Nigeria (CBN) and comply with the foreign-exchange (FX) and external-debt rules that govern all foreign borrowing by Nigerian banks.
Pre-payment is treated as any other repayment of external debt: the bank must first settle any accrued i[. . .]
Why wrong: The model *directly contradicts* the correct answer: CBN policy **prohibits** prepayment of such loans, yet the model confidently states "Yes – banks may prepay" subject to invented approval procedures. It cites specific CBN circulars, FX reporting systems, and prudential limits that are not in the source, and omits the key ground-truth point that foreign borrowings must be subordinated debt.

Case Study 3: Fabrication

Benchmark: PRBench Finance **Model:** Sonnet 4.6
Concept: Scenario analysis
Operation: Forecasting or extrapolation
Question: User: I've got 3 months of data (orders/-day split by weekday and weekend, AOV, platform commision %, wages, COGS, VAT). Can you build me a 1-month weekly financial model for scaling to UberEats/Deliveroo with £13k cash on hand? I need to see P&L, cashflow, runway, and vreak-even orders. Show me the minimum orders/AOV to clear £1,500 profit per week [. . .]
Gold answer: Solution methodology / reasoning chain: Consider the baseline assumption, and then model the optimistic scenario assuming a 30% increase in sales and a pessimistic scenario which includes a 25% decrease in sales, a 20% increase in food costs, and other provided information.
Determine the minimum number of orders that must be processed daily to bre [. . .]
Model response (excerpt): # Comprehensive Financial Model: Scaling to UberEats/Deliveroo
—
SECTION 0: DATA INPUTS & ASSUMPTIONS
Before building the model, here is a clear declaration of what I am using versus assuming. **Replace my assumptions with your actual data where indicated.**
—
YOUR PROVIDED DATA (Assumed Values — Replace With Actuals)
Since you referenced having 3 months of data but haven't shared the specific figures, I have constructed **realistic baseline inputs** for a UK home-cook scaling t [. . .]
Why wrong: The model invented nearly all core inputs (weekday/weekend orders, AOV £18.50, wages £320, packaging, fixed costs, VAT status, Monte Carlo outputs, etc.) even though the user said they had 3 months of actual data and the source only provided earlier illustrative assumptions like £18 AOV, £250/week other costs, and £2.50 delivery fee. It also ignored source-specific items such as delivery costs and [. . .]

B.3 Case Studies: Models Error in Similar Ways

Case Study 1: Convergent Reasoning Error

Benchmark: SECQUE **Concept:** Debt, leverage, and credit risk **Operation:** Ratio calculation
Convergent error: Incorrect reasoning chain

Question: What is ExxonMobil's debt ratio as of June 30, 2024?
Gold answer: Debt Ratio = Total Debt / Total Assets = (36,565 + 6,621) / 460,707 = 0.094 = **9.4%**. This low debt ratio reflects ExxonMobil's conservative approach to debt.

Claude 4.6 Sonnet: "## ExxonMobil's Debt Ratio as of June 30, 2024. The **debt ratio** is calculated as: $$\text{Debt R[...]ually **improved** (decreased) by approximately 3.5 percentage points during the first half of 2024."

Claude 4.6 Opus: "# ExxonMobil's Debt Ratio as of June 30, 2024. The **debt ratio** is calculated as: $$\text{Debt Ra[...]he significant increase in total assets (from the Pioneer Natural Resources acquisition) and equity."

GPT-5.4: "ExxonMobil's debt ratio as of June 30, 2024 is: [...] = 184,441 / 460,707 ≈ 0.400. So, the debt ratio is **about 0.40, or 40.0%**."

Nemotron 3 Super: "40.0%"

Why this is convergent: Every model chose the same incorrect formula (total liabilities ÷ total assets) rather than the specified definition (total *debt* ÷ total assets, where debt = long-term debt + notes and loans payable). The error is not arithmetic—all models computed correctly given their formula—but conceptual: they defaulted to a generic textbook "liabilities-to-assets" interpretation instead of the context-specific definition that distinguishes financial debt from operating liabilities.

Case Study 2: Convergent Value Retrieval Error

Benchmark: SECQUE **Concept:** Segment, geography, and product breakdowns **Operation:** Cross-entity comparison
Convergent error: Wrong value retrieved

Question: Compare the geographic revenue distribution of Fortinet Inc. and Rapid7 Inc. for 2023.
Gold answer: Fortinet: Americas 41%, EMEA 39%, APAC 20%. Rapid7: the increase in 2023 revenue versus 2022 comprised $65.6 M from the U.S. and $23.1 M internationally (i.e., the comparison should use the *incremental* geographic split, not the absolute split).

Claude 4.6 Sonnet: "## Geographic Revenue Distribution Comparison: Fortinet vs. Rapid7 (2023). ### Fortinet, Inc. (Total[...]ing early-stage international expansion compared to Fortinet's already established global footprint."

Claude 4.6 Opus: "# Geographic Revenue Distribution Comparison: Fortinet vs. Rapid7 (2023). ## Fortinet, Inc. (FY 2023[...]remains predominantly U.S.-focused with a growing but still relatively small international presence."

GPT-5.4: "For 2023, the two companies show different levels of geographic detail and different revenue mixes:[...]weighted toward the United States** with a smaller contribution from all other geographies combined."

Nemotron 3 Super: "Fortinet Inc.'s 2023 revenue was distributed as 41% from the Americas, 39% from EMEA, and 20% from A[...]'s 2023 revenue was distributed as 74% from the United States and 26% from Other (non-U.S.) regions."

Why this is convergent: All models default to the most salient numeric framing (absolute revenue totals) rather than the contextually appropriate one (year-over-year *change* in geographic distribution). The Rapid7 filing discloses geographic detail only for the revenue *increase*, not the full base—but every model substitutes the total revenue split, revealing a shared bias toward the most prominent numerical pattern in the source document.

Case Study 3: Convergent Value Retrieval Error

Benchmark: DocFinQA **Concept:** Operating expenses and profitability **Operation:** Percentage change
Convergent error: Wrong value retrieved

Question: What was the change in millions of vehicles from 2015 to 2016?
Gold answer: vehicles_2015 = 74; vehicles_2016 = 68; difference = 68 − 74 = **−6**.

Claude 4.6 Sonnet: `vehicles_2016 = 8638 # in millions; vehicles_2015 = 8111 # in millions; change_in_vehicles = vehicles_2016 - vehicles_2015; answer = change_in_vehicles`

Claude 4.6 Opus: `vehicles_2016 = 8638; vehicles_2015 = 8111; change_in_vehicles = vehicles_2016 - vehicles_2015; answer = change_in_vehicles`

GPT-5.4: `# Vehicles from Property, Plant and Equipment (in millions); vehicles_2016 = 8638; vehicles_2015 = 8111; answer = change_in_vehicles_from_2015_to_2016`

Nemotron 3 Super: `527`

Why this is convergent: The source document contains both a *fleet count* table (68/74 million vehicles) and a *Property, Plant & Equipment* table with dollar values for the "Vehicles" line item ($8,638 M / $8,111 M). All four models retrieve from the PP&E table (yielding 527) rather than the fleet table (yielding −6). This reveals a shared failure in table disambiguation: when multiple tables contain a column labelled "Vehicles," every model defaults to the larger, more prominent dollar-denominated table rather than the contextually correct unit-count table.